

Supervisor Localization of Timed Discrete-Event Systems under Partial Observation and Communication Delay

Renyuan Zhang¹ and Kai Cai²

Abstract

We study *supervisor localization* for timed discrete-event systems under partial observation and communication delay in the Brandin-Wonham framework. First, we employ *timed relative observability* to synthesize a partial-observation monolithic supervisor; the control actions of this supervisor include not only disabling action of prohibitable events (as that of controllable events in the untimed case) but also “clock-preempting” action of forcible events. Accordingly we decompose the supervisor into a set of partial-observation local controllers one for each prohibitable event, as well as a set of partial-observation local preemptors one for each forcible event. We prove that these local controllers and preemptors collectively achieve the same controlled behavior as the partial-observation monolithic supervisor does. Moreover, we propose channel models for inter-agent event communication and impose bounded and unbounded delay as temporal specifications. In this formulation, there exist multiple distinct observable event sets; thus we employ *timed relative coobservability* to synthesize partial-observation decentralized supervisors, and then localize these supervisors into local controllers and preemptors. The above results are illustrated by a timed workcell example.

Keywords

Timed Discrete-Event Systems; Supervisor Localization; Partial Observation, Communication Delay.

I. INTRODUCTION

In [1], [2] we developed a top-down approach, called *supervisor localization*, to the distributed control synthesis of multi-agent discrete-event systems (DES). The essence of localization is the decomposition

¹R. Zhang is with School of Automation, Northwestern Polytechnical University, China ryzhang@nwpu.edu.cn

²K. Cai is with Urban Research Plaza, Osaka City University, Japan kai.cai@eng.osaka-cu.ac.jp

of the monolithic (optimal and nonblocking) supervisor into local controllers for the individual agents. In [3] we extended supervisor localization to timed DES (TDES) in the Brandin-Wonham framework [4]; in addition to local controllers (corresponding to disabling actions), a set of local preemptors is obtained corresponding to clock-preempting actions. More recently in [5], we extended the untimed supervisor localization to the case of partial observation. In particular, we combined localization with *relative observability* [6] to first synthesize a partial-observation monolithic supervisor, and then decompose the supervisor into local controllers whose state changes are caused only by observable events.

In this paper, we further study supervisor localization for TDES under partial observation in the Brandin-Wonham framework, thereby extending both [3] and [5]. We propose to first synthesize a partial-observation monolithic supervisor using the concept of *timed relative observability* [7]. Timed relative observability is proved to be generally stronger than timed observability [8], weaker than normality [8], and closed under set union. Therefore the supremal relatively observable (and controllable) sublanguage of a given language exists and may be effectively computed [7]. Since this supremal sublanguage is observable and controllable, it may be implemented by a partial-observation (feasible and nonblocking) supervisor [8]. We then suitably extend the localization procedure in [3] to decompose the supervisor into partial-observation local controllers and local preemptors for individual agents, and prove that the derived local controlled behavior is equivalent to the monolithic one.

Moreover, we address the issue of communication delay. First, we introduce two types of channel models for inter-agent event communication. The introduced models are treated as plant components, and the requirements of bounded and unbounded delay are treated as temporal specifications imposed on the plant. In this formulation, there are multiple distinct observable event sets. This is because the occurrence of a communication event and sending that event are observable only to the sender, but not observable to the receiver; on the other hand, the receiving of a communication event is observable only to the receiver, but not observable to the sender. To deal with multiple observable event sets, we propose to employ the concept of *relative coobservability* [7], which is closed under set union, to first synthesize a set of partial-observation decentralized supervisors, and then decompose these decentralized supervisors into the respective local controllers/preemptors. Finally, we prove that the derived local controlled behavior is identical to that achieved by the partial-observation decentralized supervisors.

The contributions of this paper are threefold. First, the proposed timed supervisor localization under partial observation extends the untimed counterpart in [5]: not only is the monolithic supervisor's disabling action localized (as in the untimed case), but also its preemptive action is localized with respect to individual forcible events. Second, the proposed partial-observation supervisor localization of TDES also extends

the full-observation counterpart in [3]. Specifically, the new concepts of *partial-observation control cover* and *partial-observation preemption cover* are defined on the state set of the partial-observation supervisor; roughly speaking, the latter corresponds to the *powerset* of the full-observation supervisor's state set. In this way, in the transition structure of the resulting local controllers/preemptors, only observable events can lead to state changes. Third, in addition to partial observation, timed supervisor localization is extended to address communication delay. A novel approach employing relative coobservability [7] is proposed, which synthesizes partial-observation local controllers/preemptors tolerant of bounded and unbounded delay of event communication.

We note that distributed/decentralized supervisory control with communication delay has been extensively studied (e.g. [9]–[13]). There are two approaches mostly related to our work. The first is a verification approach, e.g. [9], [10], which first synthesizes delay-free distributed controllers, and then verifies whether the distributed controllers tolerate specified communication delay. This approach is limited to verifying the robustness of derived controllers, but does not supply a procedure to construct controllers that are able to tolerate specified communication delay. The second approach is that of synthesis, e.g. [11]–[13], which first incorporates communication delay into the plant and specification models, and then applies decentralized control methods to synthesize distributed controllers that tolerate the communication delay. In these works, *observability*, *coobservability*, *delay-coobservability* [12], or *network observability* [13] are necessary for the existence of distributed controllers. However, these observability properties are not closed under set union, and thus there generally does not exist the respective supremal sublanguage of a given language. By contrast, we employ the recently proposed relative coobservability, which is closed under set union and the supremal relatively coobservable sublanguage is effectively computable [7].

The paper is organized as follows. Section II reviews the preliminaries on the Brandin-Wonham TDES framework. Section III formulates the partial-observation supervisor localization problem of TDES, and Section IV develops the solution localization procedure. Section V investigates partial-observation supervisor localization with communication delay by using the concept of relative coobservability. Finally Section VI states our conclusions.

II. PRELIMINARIES

This section reviews supervisory control of TDES in the Brandin-Wonham framework [4], [14, Chapter 9]. First consider the untimed DES model $\mathbf{G}_{act} = (A, \Sigma_{act}, \delta_{act}, a_0, A_m)$; here A is the finite set of *activities*, Σ_{act} the finite set of *events*, $\delta_{act} : A \times \Sigma_{act} \rightarrow A$ the (partial) *transition function*, $a_0 \in A$ the *initial activity*, and $A_m \subseteq A$ the set of *marker activities*. Let \mathbb{N} denote the set of natural numbers

$\{0, 1, 2, \dots\}$, and introduce *time* into \mathbf{G}_{act} by assigning to each event $\sigma \in \Sigma_{act}$ a *lower bound* $l_\sigma \in \mathbb{N}$ and an *upper bound* $u_\sigma \in \mathbb{N} \cup \{\infty\}$, such that $l_\sigma \leq u_\sigma$. Also introduce a distinguished event, written *tick*, to represent “tick of the global clock”. Then a TDES model

$$\mathbf{G} := (Q, \Sigma, \delta, q_0, Q_m), \quad (1)$$

is constructed from \mathbf{G}_{act} (refer to [4], [14, Chapter 9] for detailed construction) such that Q is the finite set of *states*, $\Sigma := \Sigma_{act} \dot{\cup} \{tick\}$ the finite set of events, $\delta : Q \times \Sigma \rightarrow Q$ the (partial) *state transition function*, q_0 the *initial state*, and Q_m the set of *marker states*.

Let Σ^* be the set of all finite strings of elements in $\Sigma = \Sigma_{act} \dot{\cup} \{tick\}$, including the empty string ϵ . The transition function δ is extended to $\delta : Q \times \Sigma^* \rightarrow Q$ in the usual way. The *closed behavior* of \mathbf{G} is the language $L(\mathbf{G}) := \{s \in \Sigma^* | \delta(q_0, s)!\}$ and the *marked behavior* is $L_m(\mathbf{G}) := \{s \in L(\mathbf{G}) | \delta(q_0, s) \in Q_m\} \subseteq L(\mathbf{G})$. Let $K \subseteq \Sigma^*$ be a language; its *prefix closure* is $\overline{K} := \{s \in \Sigma^* | (\exists t \in \Sigma^*) st \in K\}$. K is said to be $L_m(\mathbf{G})$ -*closed* if $\overline{K} \cap L(\mathbf{G}) = L_m(\mathbf{G})$. TDES \mathbf{G} is *nonblocking* if $\overline{L_m(\mathbf{G})} = L(\mathbf{G})$.

To use TDES \mathbf{G} in (1) for supervisory control, first designate a subset of events, denoted by $\Sigma_{hib} \subseteq \Sigma_{act}$, to be the *prohibitible* events which can be disabled by an external supervisor. Next, and specific to TDES, specify a subset of *forcible* events, denoted by $\Sigma_{for} \subseteq \Sigma_{act}$, which can *preempt* the occurrence of event *tick*. Now it is convenient to define the *controllable* event set $\Sigma_c := \Sigma_{hib} \dot{\cup} \{tick\}$. The *uncontrollable* event set is $\Sigma_{uc} := \Sigma \setminus \Sigma_c$. A sublanguage $K \subseteq L_m(\mathbf{G})$ is *controllable* if, for all $s \in \overline{K}$,

$$Elig_K(s) \supseteq \begin{cases} Elig_{\mathbf{G}}(s) \cap (\Sigma_{uc} \dot{\cup} \{tick\}) & \text{if } Elig_K(s) \cap \Sigma_{for} = \emptyset, \\ Elig_{\mathbf{G}}(s) \cap \Sigma_{uc} & \text{if } Elig_K(s) \cap \Sigma_{for} \neq \emptyset, \end{cases}$$

where $Elig_K(s) := \{\sigma \in \Sigma | s\sigma \in \overline{K}\}$ is the subset of eligible events after string s .

For partial observation, Σ is partitioned into Σ_o , the subset of observable events, and Σ_{uo} , the subset of unobservable events (i.e. $\Sigma = \Sigma_o \dot{\cup} \Sigma_{uo}$). Bring in the *natural projection* $P : \Sigma^* \rightarrow \Sigma_o^*$ defined by

$$\begin{aligned} P(\epsilon) &= \epsilon; \\ P(\sigma) &= \begin{cases} \epsilon, & \text{if } \sigma \notin \Sigma_o, \\ \sigma, & \text{if } \sigma \in \Sigma_o; \end{cases} \end{aligned} \quad (2)$$

$$P(s\sigma) = P(s)P(\sigma), \quad s \in \Sigma^*, \sigma \in \Sigma.$$

As usual, P is extended to $P : Pwr(\Sigma^*) \rightarrow Pwr(\Sigma_o^*)$, where $Pwr(\cdot)$ denotes powerset. Write $P^{-1} : Pwr(\Sigma_o^*) \rightarrow Pwr(\Sigma^*)$ for the *inverse-image function* of P . A language $K \subseteq L_m(\mathbf{G})$ is *observable* if

for every pair of strings $s, s' \in \Sigma^*$ with $P(s) = P(s')$ there holds

$$(\forall \sigma \in \Sigma) s\sigma \in \overline{K}, s' \in \overline{K}, s'\sigma \in L(\mathbf{G}) \Rightarrow s'\sigma \in \overline{K}.$$

A *supervisor* V under partial observation is any map $V : P(L(\mathbf{G})) \rightarrow Pow(\Sigma)$. Then the closed-loop system is V/\mathbf{G} with closed behavior $L(V/\mathbf{G})$ and marked behavior $L_m(V/\mathbf{G})$ ($:= L(V/\mathbf{G}) \cap L_m(\mathbf{G})$) [8]. A supervisor V is *nonblocking* if $\overline{L_m(V/\mathbf{G})} = L(V/\mathbf{G})$, and *admissible* if for each $s \in L(V/\mathbf{G})$,
(i) $\Sigma_{uc} \subseteq V(P(s))$ and

$$(ii) Elig_{\mathbf{G}}(s) \cap V(P(s)) \cap \Sigma_{for} = \emptyset, \text{ tick} \in Elig_{\mathbf{G}}(s) \\ \Rightarrow \text{tick} \in V(P(s)).$$

It has been proved [8] that a nonblocking, admissible supervisory control V exists which synthesizes a (nonempty) sublanguage $K \subseteq L_m(\mathbf{G})$ such that $L_m(V/\mathbf{G}) = K$ if and only if K is observable, controllable and $L_m(\mathbf{G})$ -closed. While controllability and $L_m(\mathbf{G})$ -closedness are properties closed under set union, observability is not; consequently when K is not observable, there generally does not exist the supremal observable (controllable and $L_m(\mathbf{G})$ -closed) sublanguage of K .

Recently in [7], we proposed a new concept of *relative observability*, which is stronger than observability, but permits the existence of the supremal relatively observable sublanguage. Let $C \subseteq L_m(\mathbf{G})$. A language $K \subseteq C$ is *relatively observable* (or C -observable), if for every pair of strings $s, s' \in \Sigma^*$ with $P(s) = P(s')$ there holds

$$(\forall \sigma \in \Sigma) s\sigma \in \overline{K}, s' \in \overline{C}, s'\sigma \in L(\mathbf{G}) \Rightarrow s'\sigma \in \overline{K}. \quad (3)$$

For an arbitrary sublanguage $E \subseteq L_m(\mathbf{G})$, write $\mathcal{CO}(E)$ for the family of C -observable, controllable and $L_m(\mathbf{G})$ -closed sublanguages of E . Then $\mathcal{CO}(E)$ is nonempty (the empty language \emptyset belongs) and is closed under set union; $\mathcal{CO}(K)$ has a unique supremal element $\sup \mathcal{CO}(E)$ given by

$$\sup \mathcal{CO}(E) = \bigcup \{K \mid K \in \mathcal{CO}(E)\}$$

which may be effectively computed [6], [7]. Note that since relative observability is stronger than observability, $\sup \mathcal{CO}(E)$ is observable (controllable and $L_m(\mathbf{G})$ -closed).

III. FORMULATION OF PARTIAL-OBSERVATION SUPERVISOR LOCALIZATION PROBLEM

Let the plant \mathbf{G} be comprised of N component TDES

$$\mathbf{G}_k = (Q_k, \Sigma_k, \delta_k, q_{0,k}, Q_{m,k}), \quad k = 1, \dots, N. \quad (4)$$

Then $\mathbf{G} = \mathbf{Comp}(\mathbf{G}_1, \dots, \mathbf{G}_N)$, where \mathbf{Comp} is the composition operator defined in [14, Chapter 9] which is used to build complex TDES from simpler ones. Note that Σ_k need not be pairwise disjoint.

These agents are implicitly coupled through a specification language $E \subseteq \Sigma^*$ that imposes a constraint on the global behavior of \mathbf{G} (E may itself be the composition of multiple component specifications). For the plant \mathbf{G} and the imposed specification E , let the generator $\mathbf{SUP} = (X, \Sigma, \xi, x_0, X_m)$ be such that

$$L_m(\mathbf{SUP}) := \sup \mathcal{CO}(E \cap L_m(\mathbf{G})). \quad (5)$$

We call \mathbf{SUP} the *controllable and observable controlled behavior*. Note that \mathbf{SUP} is not a partial-observation supervisor (to be defined in the next section), which can only contain observable events as state changers. To rule out the trivial case, we assume that $L_m(\mathbf{SUP}) \neq \emptyset$.

The control actions of \mathbf{SUP} include (i) disabling prohibitable events in Σ_{hib} and (ii) preempting event *tick* via forcible events in Σ_{for} . Accordingly, the localization of \mathbf{SUP} 's control actions under partial-observation is with respect to not only each prohibitable event's disabling action (just as the untimed counterpart in [5]), but also each forcible event's preemptive action. The latter is specific to TDES, for which we introduce below the new concept of "partial-observation local preemptor".

Let $\alpha \in \Sigma_{for}$ be an arbitrary forcible event, which may or may not be observable. We say that a generator

$$\mathbf{LOC}_\alpha^P = (Y_\alpha, \Sigma_\alpha, \eta_\alpha, y_{0,\alpha}, Y_{m,\alpha}), \quad \Sigma_\alpha \subseteq \Sigma_o \cup \{\alpha, tick\}$$

is a *partial-observation local preemptor* for α if (i) \mathbf{LOC}_α^P preempts event *tick* consistently with \mathbf{SUP} , and (ii) if $\sigma \in \{\alpha, tick\}$ is unobservable, then σ -transitions are selfloops in \mathbf{LOC}_α^P , i.e. for all $y \in Y_\alpha$, $\sigma \in \Sigma_{uo}$ implies $\eta_\alpha(y, \sigma) = y$.

First, condition (i) means that for all $s \in \Sigma^*$ if $s\alpha \in L(\mathbf{SUP})$, there holds

$$P_\alpha(s).tick \in L(\mathbf{LOC}_\alpha^P), s.tick \in L(\mathbf{G}) \Leftrightarrow s.tick \in L(\mathbf{SUP}) \quad (6)$$

where $P_\alpha : \Sigma^* \rightarrow \Sigma_\alpha^*$ is the natural projection. Notation $s.tick$ means that event *tick* occurs after string s and will be used henceforth. Note that specific to TDES, only when $s\alpha \in L(\mathbf{SUP})$ can *tick*-occurrence after s be preempted by α in \mathbf{LOC}_α^P . Second, condition (ii) requires that only observable events may cause state change in \mathbf{LOC}_α^P , i.e.

$$(\forall y, y' \in Y_\alpha, \forall \sigma \in \Sigma_\alpha) \quad y' = \eta_\alpha(y, \sigma)!, y \neq y' \Rightarrow \sigma \in \Sigma_o. \quad (7)$$

Note that the event set Σ_α of \mathbf{LOC}_α^P in general satisfies

$$\{\alpha, tick\} \subseteq \Sigma_\alpha \subseteq \Sigma_o \cup \{\alpha, tick\};$$

in typical cases, both subset containments are strict. In fact, the events in $\Sigma_\alpha \setminus \{\alpha, tick\}$ are communication events that may be critical to achieve synchronization with other partial-observation local preemptors/controllers. The Σ_α is not fixed *a priori*, but will be determined as part of the localization result presented in the next section.

Next, let $\beta \in \Sigma_{hib}$ be an arbitrary prohibitable event, which may or may not be observable. A generator

$$\mathbf{LOC}_\beta^C = (Y_\beta, \Sigma_\beta, \eta_\beta, y_{0,\beta}, Y_{m,\beta}), \quad \Sigma_\beta \subseteq \Sigma_o \cup \{\beta\}$$

is a *partial-observation local controller* for β if (i) \mathbf{LOC}_β^C enables/disables the event β (and only β) consistently with **SUP**, and (ii) if β is unobservable, then β -transitions are selfloops in \mathbf{LOC}_β^C . The event set Σ_β of \mathbf{LOC}_β^C in general satisfies $\{\beta\} \subseteq \Sigma_\beta \subseteq \Sigma_o \cup \{\beta\}$; in typical cases, both subset containments are strict. Like Σ_α above, Σ_β will be generated as part of our localization result.

We are now ready to formulate the *Partial-Observation Supervisor Localization Problem*:

Construct a set of partial-observation local preemptors $\{\mathbf{LOC}_\alpha^P \mid \alpha \in \Sigma_{for}\}$ and a set of partial-observation local controllers $\{\mathbf{LOC}_\beta^C \mid \beta \in \Sigma_{hib}\}$ with

$$\begin{aligned} L(\mathbf{LOC}) := & \left(\bigcap_{\alpha \in \Sigma_{for}} P_\alpha^{-1} L(\mathbf{LOC}_\alpha^P) \right) \\ & \cap \left(\bigcap_{\beta \in \Sigma_{hib}} P_\beta^{-1} L(\mathbf{LOC}_\beta^C) \right) \end{aligned} \quad (8)$$

$$\begin{aligned} L_m(\mathbf{LOC}) := & \left(\bigcap_{\alpha \in \Sigma_{for}} P_\alpha^{-1} L_m(\mathbf{LOC}_\alpha^P) \right) \\ & \cap \left(\bigcap_{\beta \in \Sigma_{hib}} P_\beta^{-1} L_m(\mathbf{LOC}_\beta^C) \right) \end{aligned} \quad (9)$$

*such that the collective controlled behavior of **LOC** is equivalent to the controllable and observable controlled behavior **SUP** in (5) with respect to **G**, i.e.*

$$L(\mathbf{G}) \cap L(\mathbf{LOC}) = L(\mathbf{SUP}),$$

$$L_m(\mathbf{G}) \cap L_m(\mathbf{LOC}) = L_m(\mathbf{SUP}).$$

Having a set of partial-observation local preemptors, one for each forcible event, and a set of partial-observation local controllers, one for each prohibitable event, we can allocate each preemptor/controller to the agent(s) owning the corresponding forcible/prohibitable event. Thereby we build for a multi-agent DES a nonblocking distributed control architecture under partial observation.

IV. PARTIAL-OBSERVATION LOCALIZATION PROCEDURE

We solve the Partial-Observation Supervisor Localization Problem of TDES by developing a partial-observation localization procedure for the preemptive and disabling action, respectively. The procedure extends the untimed counterpart in [5]. In particular, localizing the preemption of event *tick* with respect to each forcible event under partial observation is novel in the current TDES setup, for which we introduce below the concept of “partial-observation preemption cover”.

Let $\mathbf{G} = (Q, \Sigma, \delta, q_0, Q_m)$ be the TDES plant, $\Sigma_o \subseteq \Sigma$ the subset of observable events, and $P : \Sigma^* \rightarrow \Sigma_o^*$ the corresponding natural projection. Also let $\mathbf{SUP} = (X, \Sigma, \xi, x_0, X_m)$ be controllable and observable controlled behavior (as defined in (5)). We present the localization of preemptive and disabling action in the sequel. To this end, we need the concept of *uncertainty set*.

For $s \in L(\mathbf{SUP})$, let $U(s)$ be the subset of states of \mathbf{SUP} that may be reached by some string s' that looks like s , i.e.

$$U(s) = \{x \in X \mid (\exists s' \in \Sigma^*) P(s) = P(s'), x = \xi(x_0, s')\}.$$

We call $U(s)$ the *uncertainty set* [5] of the state $\xi(x_0, s)$ associated with string s . Let $\mathcal{U}(X) := \{U(s) \subseteq X \mid s \in L(\mathbf{SUP})\}$, i.e. $\mathcal{U}(X)$ is the set of uncertainty sets of all states (associated with strings in $L(\mathbf{SUP})$) in X . The size of $\mathcal{U}(X)$ is in general $|\mathcal{U}(X)| \leq 2^{|X|}$.

The transition function associated with $\mathcal{U}(X)$ is $\hat{\xi} : \mathcal{U}(X) \times \Sigma_o \rightarrow \mathcal{U}(X)$ given by

$$\hat{\xi}(U, \sigma) = \bigcup \{\xi(x, u_1 \sigma u_2) \mid x \in U, u_1, u_2 \in \Sigma_{uo}^*\}.$$

With $\mathcal{U}(X)$ and $\hat{\xi}$, define the *partial-observation monolithic supervisor*

$$\mathbf{SUPO} = (\mathcal{U}(X), \Sigma_o, \hat{\xi}, U_0, U_m), \quad (10)$$

where $U_0 = U(\epsilon)$ and $U_m = \{U \in \mathcal{U}(X) \mid U \cap X_m \neq \emptyset\}$. It is known [14] that $L(\mathbf{SUPO}) = P(L(\mathbf{SUP}))$ and $L_m(\mathbf{SUPO}) = P(L_m(\mathbf{SUP}))$.

Now let $x \in X$ be any state and $\sigma \in \Sigma_c (= \Sigma_{hib} \dot{\cup} \{tick\})$ be a controllable event. We say that (1) σ is *enabled* at x if $\xi(x, \sigma)!$; (2) σ ($\neq tick$) is *disabled* at x if $\neg \xi(x, \sigma)!$ and

$$(\exists s \in \Sigma^*) \xi(x_0, s) = x \ \& \ \delta(q_0, s\sigma)!$$

(3) σ ($= tick$) is *preempted* at x if $\neg \xi(x, tick)!$ and

$$(\exists s \in \Sigma^*) (\exists \sigma_f \in \Sigma_{for}) \xi(x_0, s) = x \ \& \ \xi(x, \sigma_f)! \ \& \ \delta(q_0, s.tick)!$$

(4) σ is *not defined* at x if $\neg \xi(x, \sigma)!$ and

$$(\forall s \in \Sigma^*) \xi(x_0, s) = x \Rightarrow \neg \delta(q_0, s\sigma)!.$$

Lemma 1. *Given the controllable and observable controlled behavior **SUP** in (5), let $U \in \mathcal{U}(X)$, $x \in U$, and $\sigma \in \Sigma_c$. If σ is enabled at x , then for all $x' \in U$, either σ is also enabled at x' , or σ is not defined in **G**. On the other hand, if σ is disabled (resp. preempted) at x , then for all $x' \in U$, either σ is also disabled (resp. preempted) at x' , or σ is not defined in **G**.*

The proof is similar to that of Lemma 1 in [5].

A. Partial-Observation Localization of Preemptive Action

Under partial observation, the preemptive action after string $s \in L(\mathbf{SUP})$ depends not on the single state $\xi(x_0, s)$, but on the uncertainty set $U(s)$, namely a state of **SUPO**.

Fix an arbitrary forcible event $\alpha \in \Sigma_{for}$. First define $E_{tick} : \mathcal{U}(X) \rightarrow \{0, 1\}$ according to

$$(\forall U \in \mathcal{U}(X)) \ E_{tick}(U) = \begin{cases} 1, & \text{if } (\exists x \in U) \xi(x, tick)!, \\ 0, & \text{otherwise.} \end{cases}$$

Thus $E_{tick}(U) = 1$ means that *tick* is enabled at some state $x \in U$. Then by Lemma 1, at any other state $x' \in U$, *tick* is either enabled or not defined. Then define $F_\alpha : \mathcal{U}(X) \rightarrow \{0, 1\}$ according to

$$(\forall U \in \mathcal{U}(X)) \ F_\alpha(U) = \begin{cases} 1, & \text{if } (\exists x \in U) \ \xi(x, \alpha)! \ \& \ \neg \xi(x, tick)! \ \& \\ & ((\exists s \in \Sigma^*) \xi(x_0, s) = x \ \& \ \delta(q_0, s.tick)!), \\ 0, & \text{otherwise.} \end{cases}$$

Hence $F_\alpha(U) = 1$ if *tick* is preempted at some state $x \in U$, i.e. forcible event α is defined at state x , which effectively preempts the occurrence of event *tick*. Again by Lemma 1, at any other state $x' \in U$, *tick* is either preempted or not defined. Note that at state x , α need not be the only forcible event that preempts *tick*, for there can be other forcible events, say α' , defined at x . In that case, $F_{\alpha'}(U) = 1$ holds as well.

Based on the preemption information captured by E_{tick} and F_α above, we define the preemption consistency relation $\mathcal{R}_\alpha^P \subseteq \mathcal{U}(X) \times \mathcal{U}(X)$ (for α) as follows.

Definition 1. For $U, U' \in \mathcal{U}(X)$, we say that U and U' are *preemption consistent* with respect to α , written $(U, U') \in \mathcal{R}_\alpha^P$, if $E_{tick}(U) \cdot F_\alpha(U') = 0 = E_{tick}(U') \cdot F_\alpha(U)$.

Thus a pair of uncertainty sets (U, U') satisfies $(U, U') \in \mathcal{R}_\alpha^P$ if *tick* is defined at some state of U , but not preempted by α at any state of U' , and vice versa. It is easily verified that \mathcal{R}_α^P is reflexive and

symmetric, but not transitive. Hence \mathcal{R}_α^P is not an equivalence relation. This fact leads to the definition of a *partial-observation preemption cover*.

Definition 2. Let I_α be some index set, and $\mathcal{C}_\alpha^P = \{\mathcal{U}_i \subseteq \mathcal{U}(X) | i \in I_\alpha\}$ be a cover on $\mathcal{U}(X)$. We say that \mathcal{C}_α^P is a *partial-observation preemption cover* with respect to α if

- (i) $(\forall i \in I_\alpha, \forall U, U' \in \mathcal{U}_i) (U, U') \in \mathcal{R}_\alpha^P,$
- (ii) $(\forall i \in I_\alpha, \forall \sigma \in \Sigma_o)(\exists U \in \mathcal{U}_i) \hat{\xi}(U, \sigma) \neq \emptyset \Rightarrow$
 $((\exists j \in I_\alpha)(\forall U' \in \mathcal{U}_j) \hat{\xi}(U', \sigma) \neq \emptyset \Rightarrow \hat{\xi}(U', \sigma) \in \mathcal{U}_j).$

A partial-observation preemption cover \mathcal{C}_α^P lumps the uncertainty sets $U \in \mathcal{U}(X)$ into (possibly overlapping) *cells* $\mathcal{U}_i \in \mathcal{C}_\alpha^P, i \in I_\alpha$, according to (i) the uncertainty sets U that reside in the same cell \mathcal{U}_i must be pairwise preemption consistent, and (ii) for every observable event $\sigma \in \Sigma_o$, the uncertainty sets U' that can be reached from any uncertainty set $U \in \mathcal{U}_i$ by a one-step transition σ must be covered by the same cell \mathcal{U}_j . Inductively, two uncertainty sets U and U' belong to a common cell of \mathcal{C}_α^P if and only if U and U' are preemption consistent, and two future uncertainty sets that can be reached respectively from U and U' by a given observable string are again preemption consistent.

The partial-observation preemption cover \mathcal{C}_α^P differs from its full-observation counterpart in [3] in two aspects. First, \mathcal{C}_α^P is defined on $\mathcal{U}(X)$, not on X ; this is due to state uncertainty caused by partial observation. Second, in condition (ii) of \mathcal{C}_α^P only observable events in Σ_o are considered, not Σ ; this is to generate partial-observation local preemptors whose state transitions are triggered only by observable events. We call \mathcal{C}_α^P a *partial-observation preemption congruence* if \mathcal{C}_α^P happens to be a partition on $\mathcal{U}(X)$.

Having defined a partial-observation preemption cover \mathcal{C}_α^P on $\mathcal{U}(X)$, we construct a generator $\mathbf{J}_\alpha = (I_\alpha, \Sigma_o, \zeta_\alpha, i_{0,\alpha}, I_{m,\alpha})$ and two functions $\psi_\alpha : I_\alpha \rightarrow \{0, 1\}$ and $\psi_{tick} : I_\alpha \rightarrow \{0, 1\}$ as follows:

$$(i) \quad i_{0,\alpha} \in I_\alpha \text{ such that } (\exists U \in \mathcal{U}_{i_{0,\alpha}}) x_0 \in U; \quad (11)$$

$$(ii) \quad I_{m,\alpha} := \{i \in I_\alpha | (\exists U \in \mathcal{U}_i) X_m \cap U \neq \emptyset\}; \quad (12)$$

$$(iii) \quad \zeta_\alpha : I_\alpha \times \Sigma_o \rightarrow I_\alpha \text{ with } \zeta_\alpha(i, \sigma) = j \\ \text{if } (\exists U \in \mathcal{U}_i) \hat{\xi}(U, \sigma) \in \mathcal{U}_j; \quad (13)$$

$$(iv) \quad \psi_\alpha(i) = 1 \text{ iff } (\exists U \in \mathcal{U}_i)(\exists x \in U) \xi(x, \alpha)!. \quad (14)$$

$$(v) \quad \psi_{tick}(i) = 1 \text{ iff } (\exists U \in \mathcal{U}_i) E_{tick}(U) = 1. \quad (15)$$

The function $\psi_\alpha(i) = 1$ means that forcible event α is defined at state i of \mathbf{J}_α , and the function $\psi_{tick}(i) = 1$ means that event $tick$ is enabled at state i of \mathbf{J}_α . Note that owing to cell overlapping, the choices of $i_{0,\alpha}$ and ζ_α may not be unique, and consequently \mathbf{J}_α may not be unique. In that case we simply pick an arbitrary instance of \mathbf{J}_α .

Finally we define the *partial-observation local preemptor* $\mathbf{LOC}_\alpha^P = (Y_\alpha, \Sigma_\alpha, \eta_\alpha, y_{0,\alpha}, Y_{m,\alpha})$ as follows:
 Step (i) $Y_\alpha = I_\alpha$, $y_{0,\alpha} = i_{0,\alpha}$, and $Y_{m,\alpha} = I_{m,\alpha}$. Thus the function ψ_α is $\psi_\alpha : Y_\alpha \rightarrow \{0, 1\}$, and the function ψ_{tick} is $\psi_{tick} : Y_\alpha \rightarrow \{0, 1\}$.

Step (ii) $\Sigma_\alpha = \{\alpha, tick\} \cup \Sigma_{com,\alpha}$, where

$$\begin{aligned} \Sigma_{com,\alpha} := \{ \sigma \in \Sigma_o \setminus \{\alpha, tick\} \mid (\exists i, j \in I_\alpha) i \neq j \ \& \\ \zeta_\alpha(i, \sigma) = j \} \end{aligned} \quad (16)$$

Thus $\Sigma_{com,\alpha}$ is the set of observable events that are not merely selfloops in \mathbf{J}_α . It holds by definition that $\{\alpha, tick\} \subseteq \Sigma_\alpha \subseteq \Sigma_o \cup \{\alpha, tick\}$, and $\Sigma_{com,\alpha}$ contains the events of other local controllers that need to be communicated to \mathbf{LOC}_α .

Step (iii) If $\alpha \in \Sigma_o$, then $\eta_\alpha = \zeta_\alpha|_{Y_\alpha \times \Sigma_\alpha} : Y_\alpha \times \Sigma_\alpha \rightarrow Y_\alpha$, i.e. η_α is the restriction of ζ_α to $Y_\alpha \times \Sigma_\alpha$. If $\alpha \in \Sigma_{uo}$, first obtain $\eta_\alpha = \zeta_\alpha|_{Y_\alpha \times \Sigma_\alpha}$, then add α -selfloops $\eta_\alpha(y, \alpha) = y$ to those $y \in Y_\alpha$ with $\psi_\alpha(y) = 1$.
 Step (iv) If $tick \in \Sigma_{uo}$, then add $tick$ -selfloops $\eta_\alpha(y, tick) = y$ to those $y \in Y_\alpha$ with $\psi_{tick}(y) = 1$.

Lemma 2. *The generator \mathbf{LOC}_α^P is a partial-observation local preemptor for α , i.e. (6) and (7) hold.*

We postpone the proof of Lemma 2 after our main result, Theorem 1, in subsection IV-C.

By the same procedure, we generate a set of partial-observation local preemptors \mathbf{LOC}_α^P , one for each forcible event $\alpha \in \Sigma_{for}$. We will verify below that these generated preemptors collectively achieve the same *tick*-preemptive action as **SUP** did.

B. Partial-Observation Localization of Disabling Action

Next, we turn to the localization of disabling action, which is analogous to the treatment in [5] for the untimed case. Fix an arbitrary prohibitible event $\beta \in \Sigma_{hib}$. Define $E_\beta : \mathcal{U}(X) \rightarrow \{0, 1\}$ according to

$$(\forall U \in \mathcal{U}(X)) \ E_\beta(U) \text{ iff } (\exists x \in U) \xi(x, \beta)!$$

So $E_\beta(U) = 1$ if event α is enabled at some state $x \in U$. Also define $D_\beta : \mathcal{U}(X) \rightarrow \{0, 1\}$ according to

$$D_\beta(U) = \begin{cases} 1, & \text{if } (\exists x \in U) \neg \xi(x, \beta)! \ \& \\ & ((\exists s \in \Sigma^*) \xi(x_0, s) = x \ \& \ \delta(q_0, s\beta)!), \\ 0, & \text{otherwise.} \end{cases}$$

Hence $D_\beta(U) = 1$ if β is disabled at some state $x \in U$. Now define $M : \mathcal{U}(X) \rightarrow \{0, 1\}$ by $M(U) = 1$ iff there exists $x \in U$ such that $x \in X_m$; and $T : \mathcal{U}(X) \rightarrow \{0, 1\}$ by $T(U) = 1$ iff there exists $s \in \Sigma^*$ such that $\xi(x_0, s) \in U$ and $\delta(q_0, s) \in Q_m$.

We define the *control consistency relation* $\mathcal{R}_\beta^C \subseteq \mathcal{U}(X) \times \mathcal{U}(X)$ with respect to β according to $(U, U') \in \mathcal{R}_\beta^C$ iff

$$E_\beta(U) \cdot D_\beta(U') = 0 = E_\beta(U') \cdot D_\beta(U)$$

$$T(U) = T(U') \Rightarrow M(U) = M(U').$$

Let I_β be some index set, and $\mathcal{C}_\beta^C = \{\mathcal{U}_i \subseteq \mathcal{U}(X) | i \in I_\beta\}$ a cover on $\mathcal{U}(X)$. We say that \mathcal{C}_β^C is a *partial-observation control cover* with respect to β if

$$\begin{aligned} (i) \quad & (\forall i \in I_\beta, \forall U, U' \in \mathcal{U}_i) (U, U') \in \mathcal{R}_\beta^C, \\ (ii) \quad & (\forall i \in I_\beta, \forall \sigma \in \Sigma_o) (\exists U \in \mathcal{U}_i) \hat{\xi}(U, \sigma) \neq \emptyset \Rightarrow \\ & ((\exists j \in I_\beta) (\forall U' \in \mathcal{U}_j) \hat{\xi}(U', \sigma) \neq \emptyset \Rightarrow \hat{\xi}(U', \sigma) \in \mathcal{U}_j). \end{aligned}$$

With the control cover \mathcal{C}_β^C on $\mathcal{U}(X)$, we construct, by the Steps (i)-(iii) above for a local preemptor, a partial-observation local controller $\mathbf{LOC}_\beta^C = (Y_\beta, \Sigma_\beta, \eta_\beta, y_{0,\beta}, Y_{m,\beta})$ for prohibitable event β . Here, the event set Σ_β is $\Sigma_\beta = \{\beta\} \cup \Sigma_{com,\beta}$, where

$$\Sigma_{com,\beta} := \{\sigma \in \Sigma_o \setminus \{\beta\} \mid (\exists i, j \in I_\beta) i \neq j, \zeta_\alpha(i, \sigma) = j\}. \quad (17)$$

Lemma 3. *The generator \mathbf{LOC}_β^C is a partial-observation local controller for prohibitable event β .*

For a proof of Lemma 3, see [5, Lemma 2].

By the same procedure, we generate a set of partial-observation local controllers \mathbf{LOC}_β^C , one for each prohibitable event $\beta \in \Sigma_{hib}$. We will verify below that these generated controllers collectively achieve the same disabling action as **SUP** did.

C. Main Result

Here is the main result of this section, which states that the collective behavior of the partial-observation local preemptors and local controllers generated by the localization procedure above is identical to the monolithic controllable and observable **SUP**.

Theorem 1. *The set of partial-observation local preemptors $\{\mathbf{LOC}_\alpha^P | \alpha \in \Sigma_{hib}\}$ and the set of partial-observation local controllers $\{\mathbf{LOC}_\alpha^C | \beta \in \Sigma_{hib}\}$ constructed above solve the Partial-Observation Supervisor Localization Problem, i.e.*

$$L(\mathbf{G}) \cap L(\mathbf{LOC}) = L(\mathbf{SUP}) \quad (18)$$

$$L_m(\mathbf{G}) \cap L_m(\mathbf{LOC}) = L_m(\mathbf{SUP}) \quad (19)$$

where $L(\mathbf{LOC})$ and $L_m(\mathbf{LOC})$ are as defined in (8) and (9), respectively.

Since for every partial-observation preemption cover (resp. control cover), the presented procedure constructs a local preemptor (resp. local controller), Theorem 1 asserts that every set of preemption and control covers together generates a solution to the Partial-Observation Supervisor Localization Problem. The localization algorithm in [5] for untimed DES can easily be adapted in the current TDES case, the only modification being to use the new definitions of partial-observation preemption and control consistency given in Sections IV-A and IV-B. The complexity of the localization algorithm is $O(n^4)$; since the size n of $\mathcal{U}(X)$ is $n \leq 2^{|X|}$ in general, the algorithm is exponential in $|X|$.

Proof of Theorem 1: First, we prove (\subseteq) of (18), i.e. $L(\mathbf{G}) \cap L(\mathbf{LOC}) \subseteq L(\mathbf{SUP})$, by induction on the length of strings.

For the **base step**, note that none of $L(\mathbf{G})$, $L(\mathbf{LOC})$ and $L(\mathbf{SUP})$ is empty; and thus the empty string ϵ belongs to all of them. For the **inductive step**, suppose that $s \in L(\mathbf{G}) \cap L(\mathbf{LOC})$, $s \in L(\mathbf{SUP})$ and $s\sigma \in L(\mathbf{G}) \cap L(\mathbf{LOC})$ for arbitrary event $\sigma \in \Sigma$; we must show that $s\sigma \in L(\mathbf{SUP})$. Since $\Sigma = \Sigma_{uc} \dot{\cup} \Sigma_{hib} \dot{\cup} \{tick\}$, we consider the following three cases.

(1) $\sigma \in \Sigma_{uc}$. Since $L(\mathbf{SUP})$ is controllable, and $s\sigma \in L(\mathbf{G})$ (i.e. $\sigma \in \text{Elig}_{\mathbf{G}}(s)$), we have $\sigma \in \text{Elig}_{L_m(\mathbf{SUP})}(s)$. That is, $s\sigma \in \overline{L_m(\mathbf{SUP})} = L(\mathbf{SUP})$.

(2) $\sigma = tick$. By the hypothesis that $s, s.tick \in L(\mathbf{LOC})$, for every forcible event $\alpha \in \Sigma_{for}$, $s, s.tick \in P_\alpha^{-1}L(\mathbf{LOC}_\alpha^P)$, i.e. $P_\alpha(s), P_\alpha(s).tick \in L(\mathbf{LOC}_\alpha^P)$. Let $y = \eta_\alpha(y_{0,\alpha}, P_\alpha(s))$; then $\eta_\alpha(y, tick)!$. Since $tick$ may be observable or unobservable, we consider the following two cases.

(2.1) $tick \in \Sigma_{uo}$. It follows from the construction rule (iv) of \mathbf{LOC}_α^P that $\eta_\alpha(y, tick)$ implies that for the state $i \in I$ of the generator \mathbf{J}_α corresponding to y (i.e. $i = \zeta_\alpha(i_0, P(s))$), there holds $\psi_{tick}(i) = 1$. By the

definition of ψ_{tick} in (15), there exists an uncertainty set $U \in \mathcal{U}_i$ such that $E_{tick}(U) = 1$. Let $\xi(x_0, s) \in U'$; then $U' \in \mathcal{U}_i$. Since U and U' belong to the same cell \mathcal{U}_i , by the definition of partial-observation preemption cover they must be preemption consistent, i.e. $(U, U') \in \mathcal{R}_\alpha^P$. Thus $E_{tick}(U) \cdot F_\alpha(U') = 0$, which implies that $F_\alpha(U') = 0$. The latter means that for all state $x \in U'$, (i) $\neg\xi(x, \alpha)!$, (ii) $\xi(x, tick)!$, or (iii) $(\neg\exists s \in \Sigma^*) (\xi(x_0, s) = x \text{ and } \delta(q_0, s.tick)!)!$. First, Case (iii) is impossible for $\xi(x_0, s)$, because by hypothesis that $s \in L(\mathbf{SUP})$ and $s.tick \in L(\mathbf{G})$, we have $\xi(x_0, s)!$ and $\delta(q_0, s.tick)!$. Next, Case (ii) means directly that $s.tick \in L(\mathbf{SUP})$. Finally, Case (i) implies that $\alpha \notin Elig_{L_m(\mathbf{SUP})}(s)$; note that this holds for all $\beta \in \Sigma_{for}$. Hence $Elig_{L_m(\mathbf{SUP})}(s) \cap \Sigma_{for} = \emptyset$. Then by the fact that $L_m(\mathbf{SUP})$ is controllable and $s.tick \in L(\mathbf{G})$, $tick \in Elig_{L_m(\mathbf{SUP})}(s)$, i.e. $s.tick \in L(\mathbf{SUP})$.

(2.2) $tick \in \Sigma_o$. In this case, for the state $i \in I$ of the generator \mathbf{J}_α corresponding to y (i.e. $i = \zeta_\alpha(i_0, P(s))$), there holds $\zeta_\alpha(i, tick)!$. By the definition of ζ_α in (13), there exists an uncertainty set $U \in \mathcal{U}_i$ such that $\hat{\xi}(U, tick)!$. So $E_{tick}(U) = 1$. The rest of the proof is identical to Case (2.1) above, and we conclude that $s.tick \in L(\mathbf{SUP})$ as well.

(3) $\sigma \in \Sigma_{hib}$. There must exist a partial-observation local controller \mathbf{LOC}_σ^C for σ . It follows from $s\sigma \in L(\mathbf{LOC})$ that $s\sigma \in P_\sigma^{-1}L(\mathbf{LOC}_\sigma^C)$ and $s \in P_\sigma^{-1}L(\mathbf{LOC}_\sigma^C)$. So $P_\sigma(s\sigma) \in L(\mathbf{LOC}_\sigma^C)$ and $P_\sigma(s) \in L(\mathbf{LOC}_\sigma^C)$, namely, $\eta_\sigma(y_0, P_\sigma(s\sigma))!$ and $\eta_\sigma(y_0, P_\sigma(s))!$. Let $y := \eta_\sigma(y_0, P_\sigma(s))$; then $\eta_\sigma(y, \sigma)!$ (because $\sigma \in \Sigma_\sigma$). Since σ may be observable or unobservable, we consider the following two cases.

(3.1) $\sigma \in \Sigma_{uo}$. It follows from the construction rules of \mathbf{LOC}_σ that $\eta_\sigma(y, \sigma)!$ implies that for the state $i \in I$ of the generator \mathbf{J}_σ corresponding to y (i.e. $i = \zeta_\sigma(i_0, P(s))$), there holds $\psi_\sigma(i) = 1$. By the definition of ψ_σ in (14), there exists an uncertainty set $U \in \mathcal{U}_i$ such that $E_\alpha(U) = 1$. Let $\xi(x_0, s) \in U'$; then $U' \in \mathcal{U}_i$. Since U and U' belong to the same cell \mathcal{U}_i , by the definition of partial-observation control cover they must be control consistent, i.e. $(U, U') \in \mathcal{R}_\sigma^C$. Thus $E_\sigma(U) \cdot D_\sigma(U') = 0$, which implies $D_\sigma(U') = 0$. The latter means that for all states $x \in U'$, either (i) $\xi(x, \sigma)!$ or (ii) for all $t \in \Sigma^*$ with $\xi(x_0, t) = x$, $\delta(q_0, t\sigma)$ is not defined. Note that (ii) is impossible for $\xi(x_0, s) \in U'$, because $s\sigma \in L(\mathbf{G})$. Thus by (i), $\xi(\xi(x_0, s), \sigma)!$, and therefore $s\sigma \in L(\mathbf{SUP})$.

Case (3.2) $\sigma \in \Sigma_o$. In this case, for the state $i \in I$ of the generator \mathbf{J}_σ corresponding to y (i.e. $i = \zeta_\sigma(i_0, P(s))$), there holds $\zeta_\sigma(i, \sigma)!$. By the definition of ζ_σ in (13), there exists an uncertainty set $U \in \mathcal{U}_i$ such that $\hat{\xi}(U, \sigma)!$, i.e. $E_\sigma(U) = 1$. The rest of the proof is identical to Case (3.1) above, and we conclude that $s\sigma \in L(\mathbf{SUP})$ in this case as well.

The (\supseteq) direct of (18), as well as equation (19) can be established similarly to [5].

Finally, we provide the proof of Lemma 2.

Proof of Lemma 2. We must prove (6) and (7).

First, for (\Rightarrow) of Eq. (6), let $P_\alpha(s).tick \in L(\mathbf{LOC}_\alpha^P)$, $s.tick \in L(\mathbf{G})$ and $s\alpha \in L(\mathbf{SUP})$; we must prove that $s.tick \in L(\mathbf{SUP})$. Since $s \in L(\mathbf{SUP})$, we have $s \in P_\alpha^{-1}L(\mathbf{LOC}_\alpha)$, and thus $P_\alpha(s) \in L(\mathbf{LOC}_\alpha^P)$. Let $y := \eta_\alpha(y_0, P_\alpha(s))!$; by $P_\alpha(s).tick \in L(\mathbf{LOC}_\alpha^P)$, $\eta_\alpha(y, tick)!$. The rest of the proof is identical to the inductive case of proving (\subseteq) of (18), and we conclude that $s.tick \in L(\mathbf{SUP})$.

Next, for (\Leftarrow) of Eq. (6), let $s.tick \in L(\mathbf{SUP})$ and $s\alpha \in L(\mathbf{SUP})$; $s \in L(\mathbf{SUP})$ and $s.tick \in L(\mathbf{G})$ are immediate, and it is left to show that $P_\alpha(s).tick \in L(\mathbf{LOC}_\alpha^P)$. By $s.tick \in L(\mathbf{SUP})$ and (18), we have for all $\sigma \in \Sigma_{for}$, $s.tick \in P_\sigma^{-1}L(\mathbf{LOC}_\sigma^P)$. Because $\alpha \in \Sigma_{hib}$, we have $s.tick \in P_\alpha^{-1}L(\mathbf{LOC}_\alpha^P)$, and thus $P_\alpha(s.tick) \in L(\mathbf{LOC}_\alpha^P)$. According to the definition of Σ_α , $\{tick\} \subseteq \Sigma_\alpha$. Hence, $P_\alpha(s).tick = P_\alpha(s.tick) \in L(\mathbf{LOC}_\alpha^P)$.

Finally, to prove (7), let $y, y' \in Y_\alpha$ and $\sigma \in \Sigma_o$ and assume that $y' = \eta_\alpha(y, \sigma)$ and $y \neq y'$; we prove that $\sigma \in \Sigma_o$ by contradiction. Suppose that $\sigma \in \Sigma_{uo}$. According to (13), for all $i \in I$, $\zeta_\alpha(i, \sigma)$ is not defined. Further, according to the rules (iii) and (iv) of constructing \mathbf{LOC}_α^P , (1) for all $y \in Y$, $\eta_\alpha(y, \sigma)$ is not defined, contradicting to the assumption that $y' = \eta_\alpha(y, \alpha)$; (2) the selfloop $\eta_\alpha(y, \alpha) = y$ is added to η_α when $\psi_\alpha(y) = 1$ or $\psi_{tick}(y) = 1$, which, however, contradicts the assumption that $y \neq y'$. So we conclude that $\sigma \in \Sigma_o$.

□

For illustration, the proposed partial-observation supervisor localization procedure for TDES is applied to a case study on a timed workcell adapted from [14] in the next subsection.

D. Case Study: Timed Workcell

We illustrate partial-observation supervisor localization in TDES by studying a timed workcell example, taken from [14, Chapter 9]. As displayed in Fig. 1, the workcell consists of two machines **M1** and **M2** linked by a one-slot buffer **BUF**. The untimed DES models of the machines are displayed in Fig. 2. The workcell operates as follows. Initially the buffer is empty. With the event α_1 , **M1** takes a workpiece from the infinite workpiece source. Subsequently **M1** either breaks down (event λ_1), or successfully completes its work cycle, deposits the workpiece in the buffer (event β_1). **M2** operates similarly, but takes its workpiece from the buffer (event α_2), and deposits it when finished in the infinite workpiece sink. If a machine **Mi** ($i = 1, 2$) breaks down (event λ_i), then it will be started to repair (event μ_i), and finally its repair will be finished (event η_i). Assign lower and upper time bounds to each event, with notation (event, lower bound, upper bound), as follows:

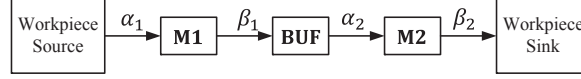


Fig. 1. Workcell: system configuration

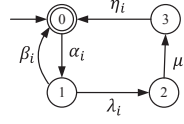


Fig. 2. Untimed DES models of M_i

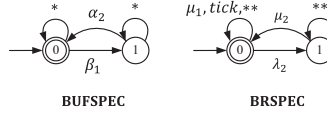


Fig. 3. Control specifications: $*$ = $\{tick, \alpha_1, \lambda_1, \mu_1, \eta_1, \beta_2, \lambda_2, \mu_2, \eta_2\}$, and $**$ = $\{\alpha_1, \beta_1, \lambda_1, \eta_1, \alpha_2, \beta_2, \eta_2\}$

M1's timed events :

$$(\alpha_1, 0, \infty) \quad (\beta_1, 1, 2) \quad (\lambda_1, 0, 2) \quad (\mu_1, 0, \infty) \quad (\eta_1, 1, \infty)$$

M2's timed events :

$$(\alpha_2, 0, \infty) \quad (\beta_2, 1, 1) \quad (\lambda_2, 0, 1) \quad (\mu_2, 0, \infty) \quad (\eta_2, 2, \infty)$$

So α_i , μ_i and η_i , $i = 1, 2$ are remote events (upper bound ∞), and β_i and λ_i , $i = 1, 2$ are prospective events (finite upper bounds). Now the TDES models of the two machine can be generated [14]; their joint behavior is the composition of the two TDES, which is the plant to be controlled, i.e. **PLANT** = **Comp**(**M1**, **M2**).

To impose behavioral constraints on the two machine's joint behavior, we take $\Sigma_{for} = \Sigma_{hib} = \{\alpha_i, \mu_i | i = 1, 2\}$, and $\Sigma_{uc} = \{\beta_i, \lambda_i, \eta_i | i = 1, 2\}$. We impose the following control specifications: (S1) **BUF** must not overflow or underflow; (S2) if **M2** goes down, its repair must be started “immediately”, and prior to starting repair of **M1** if **M1** is currently down. These two specifications are formalized as generators **BUFSPEC** and **BRSPEC**, as represented in Fig. 3. So the overall specification imposed on the **PLANT** is represented by **SPEC** = **BUFSPEC**||**BRSPEC**.

Under partial observation, we consider the case that the subset of unobservable events $\Sigma_{uo} = \{\alpha_1, \lambda_1, \mu_1, \eta_1, \beta_2\}$.

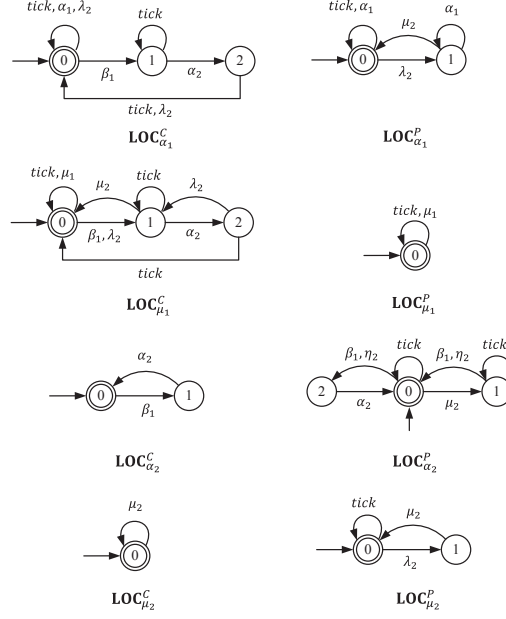


Fig. 4. Local preemptors and local controller under partial observation P ($\Sigma_{uo} = \{\alpha_1, \lambda_1, \mu_1, \eta_1, \beta_2\}$)

Namely, in **M1**, only event β_1 is observable, and in **M2**, only event β_2 is unobservable. We first compute as in (5) the controllable and observable controlled behavior **SUP**, which has 69 states and 139 transitions. Then we apply partial-observation supervisor localization to construct partial-observation local preemptors and partial-observation local controllers, respective for each forcible event and each prohibitable event. The computation can be down by an algorithm adapted from [5], as discussed in Section IV.C. The results are displayed in Fig. 4. It is verified that the collective controlled behavior of these local preemptors and controllers is equivalent to that represented by **SUP**.

$\text{LOC}_{\alpha_1}^C$ will disable event α_1 to prevent the overflow of the buffer. $\text{LOC}_{\alpha_1}^P$ may preempt the occurrence event $tick$ by α_1 when **M2** is broken down (in this case, the buffer is empty and thus α_1 may occur).

$\text{LOC}_{\mu_1}^C$ will disable event μ_1 when **M2** is broken down, as required by specification (S2). The occurrence of event μ_1 will not preempt event $tick$ (the repair of **M1** has lower priority than that of **M2**) and thus $\text{LOC}_{\mu_1}^P$ has only one state.

$\text{LOC}_{\alpha_2}^C$ will enable α_2 (**M2** get a workpiece from the buffer) when the buffer is full, i.e. **M1** has put a workpiece into the buffer. When the buffer is full and **M1** has taken a workpiece from the source, **M2** will effectively (preempt the occurrence of $tick$ by $\text{LOC}_{\alpha_2}^P$) take a workpiece from the buffer, to prevent the overflow of the buffer.

$\text{LOC}_{\mu_2}^C$ always enable μ_2 (start to repair **M2**) if it is eligible to occur, as required by specification

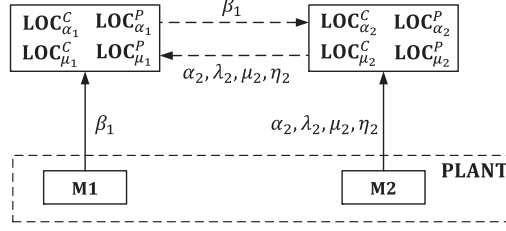


Fig. 5. Distributed control architecture: $\Sigma_o = \{\beta_1, \alpha_2, \lambda_2, \mu_2, \eta_2\}$

(S2). The repair of **M2** must be started immediately if it is enabled, thus *tick* will be preempted by $\text{LOC}_{\mu_2}^P$ after **M2** has broken down.

Finally, we allocate each partial-observation local preemptor/controller to the agent owning the corresponding forcible/prohibitible event, and build a distributed control architecture for this workcell, as displayed in Fig. 5. A local preemptor/controller either directly observes an observable event generated by the agent owning it, as denoted by solid lines in Fig. 5, or imports an observable event by communication from other local preemptors/controllers, as denoted by the dashed lines. Note that only observable events lead to state changes in the transition diagrams displayed in Fig. 4, and only the events that lead to state changes are communicated.

V. PARTIAL-OBSERVATION LOCALIZATION OF TDES WITH COMMUNICATION DELAY

As illustrated in Fig. 5, a local preemptor/controller may either directly observe an event generated by the agent owning it, or import an event by communication from other local preemptors/controllers. From now on, we address the issue of communication delay: the event communications among the plant components and their local controllers/preemptors are subject to inneglectable delays.

A. Communication Channel Models

By the supervisor localization procedure in Section IV, we obtained a set of partial-observation local preemptors LOC_{α}^P ($\alpha \in \Sigma_{for}$) with communication event set $\Sigma_{com,\alpha}$ (as in (16)), and a set of partial-observation local controllers LOC_{β}^C ($\beta \in \Sigma_{hib}$) with $\Sigma_{com,\beta}$ (as in (17)). For each event in these communication sets, consider that it is transmitted through a communication channel and subject to delay.

Let \mathbf{G}_l ($l \in \{1, \dots, N\}$) be an agent with Σ_l , and denote by $\Sigma_{com,l}$ the subset of events to be

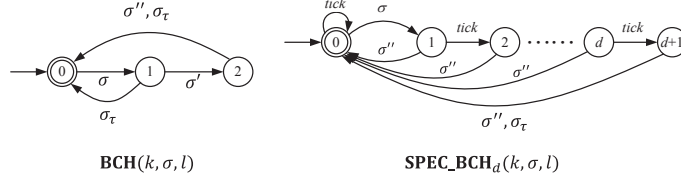


Fig. 6. Untimed DES model of bounded communication channel $\mathbf{BCH}(k, \sigma, l)$ and specification $\mathbf{SPEC_BCH}_d(k, \sigma, l)$ of delay bound d imposed on the channel.

communicated to \mathbf{G}_l which is given by

$$\Sigma_{com,l} = \left(\bigcup_{\alpha \in \Sigma_l \cap \Sigma_{for}} (\Sigma_{com,\alpha} \setminus \Sigma_l) \right) \cup \left(\bigcup_{\beta \in \Sigma_l \cap \Sigma_{hib}} (\Sigma_{com,\beta} \setminus \Sigma_l) \right).$$

Let \mathbf{G}_k ($k \in \{1, \dots, N\}$) be another agent with Σ_k . Then the subset of events communicated from agent \mathbf{G}_k to \mathbf{G}_l is

$$\Sigma_{k,com,l} = \Sigma_k \cap \Sigma_{com,l}. \quad (20)$$

Fix an event $\sigma \in \Sigma_{k,com,l}$ and consider communication delay of σ . To meet a hard deadline of an operation or to ensure system's timely performance in practice, it may often be the case that the communication delay of event σ is bounded by $d \in \mathbb{N}$ ticks. For this, we propose a channel model $\mathbf{BCH}(k, \sigma, l)$, as displayed in Fig. 6; the model will be treated as a plant component.

We explain the model $\mathbf{BCH}(k, \sigma, l)$ as follows. (1) Event σ denotes that σ occurs in \mathbf{G}_k and is sent to the communication channel. (2) Event σ' denotes that σ is received by \mathbf{G}_l , and an acknowledgement message is sent back to the channel. (3) Event σ'' denotes that \mathbf{G}_k receives the acknowledgement, which will reset the channel to be idle (i.e. the channel is ready to send the next occurrence of σ). (4) Event σ_τ denotes "timeout", which will reset the channel to idle if the transmission has not been completed in a given time $\tau = d$. The lower bounds of σ' and σ'' are both set to be 0 and the upper bounds to be d ; the lower and upper time bounds of σ_τ are both τ . The requirement of delay bound d is represented by the TDES $\mathbf{SPEC_BCH}_d(k, \sigma, l)$ displayed in Fig. 6. It is required that the time between \mathbf{G}_k sends out event σ and it receives acknowledgement σ'' be no more than d ticks. $\mathbf{SPEC_BCH}_d(k, \sigma, l)$ will be imposed as a temporal specification.

In case there happens to be no specific deadline requirement on transmission of event $\sigma \in \Sigma_{k,com,l}$, or simply no *a priori* knowledge is available of a delay bound on σ , it may be reasonable to consider

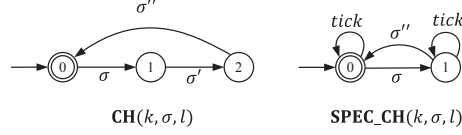


Fig. 7. Untimed DES model of unbounded communication channel $\mathbf{CH}(k, \sigma, l)$ and specification $\mathbf{SPEC_CH}(k, \sigma, l)$ of unbounded delay imposed on the channel.

unbounded delay of σ -communication. This means that the transmission of σ may take *indefinite* time to complete, although it will complete eventually. For unbounded delay, we propose a channel model $\mathbf{CH}(k, \sigma, l)$, as displayed in Fig. 7; the model will be treated as a plant component.

In $\mathbf{CH}(k, \sigma, l)$, the meaning of the event labels σ , σ' and σ'' are the same as those in bounded channel $\mathbf{BCH}(k, \sigma, l)$. In this case, however, since the communication delay is unbounded, σ' and σ'' both have lower bound 0 and upper bound ∞ (i.e. they may occur at any time after they become eligible to). The requirement of unbounded delay is represented by the TDES $\mathbf{SPEC_CH}(k, \sigma, l)$ displayed in Fig. 7; $\mathbf{SPEC_CH}(k, \sigma, l)$ will be imposed as a temporal specification.

In the channel models above, we make the following choices. (1) Both events σ' and σ'' are uncontrollable, because it is not reasonable (if not impossible) to disable the receipt of a communication or an acknowledgement; (2) event σ_τ is uncontrollable but forcible, because its occurrence is supposed to preempt *tick*; (3) events σ , σ'' (and σ_τ) are observable to the sender \mathbf{G}_k but unobservable to the receiver \mathbf{G}_l , while σ' is observable to \mathbf{G}_l but unobservable to \mathbf{G}_k . This means that the agents $\mathbf{G}_1, \dots, \mathbf{G}_N$ generally have different subsets of observable events; this is a new feature of the current formulation with communication delay.

B. Partial-Observation Supervisor Localization with Communication Delay

Recall from Section III that, in the delay-free case, we had plant $\mathbf{G} = \mathbf{Comp}(\mathbf{G}_1, \dots, \mathbf{G}_N)$ over Σ , specification $E \subseteq \Sigma^*$, prohibitable event set Σ_{hib} , and forcible event set Σ_{for} . For simplicity of presentation, we assume here that the component agents \mathbf{G}_k (given in (4)) have pairwise disjoint event sets Σ_k ($k \in \mathcal{N} := \{1, \dots, N\}$); the more general case of event sharing among agents is presented in [15].

Now for $k, l \in \mathcal{N}$ let $\Sigma_{k,com,l}$ in (20) be partitioned as $\Sigma_{k,com,l} = \Sigma_{k,com,l}^{bd} \dot{\cup} \Sigma_{k,com,l}^{ud}$, where $\Sigma_{k,com,l}^{bd}$ is the subset of communication events with bounded delay and $\Sigma_{k,com,l}^{ud}$ the subset of those with unbounded

delay. First, the new plant $\tilde{\mathbf{G}}$ including both the agents and the channels is

$$\tilde{\mathbf{G}} = \mathbf{Comp}(\mathbf{G}, \{\mathbf{BCH}(k, \sigma, l) \mid \sigma \in \Sigma_{k,com,l}^{bd}, k, l \in \mathcal{N}\}, \\ \{\mathbf{CH}(k, \sigma, l) \mid \sigma \in \Sigma_{k,com,l}^{ud}, k, l \in \mathcal{N}\}).$$

The event set $\tilde{\Sigma}$ of $\tilde{\mathbf{G}}$ is $\tilde{\Sigma} = \Sigma \cup \{\sigma', \sigma'', \sigma_\tau \mid \sigma \in \Sigma_{k,com,l}^{bd}, k, l \in \mathcal{N}\} \cup \{\sigma', \sigma'' \mid \sigma \in \Sigma_{k,com,l}^{ud}, k, l \in \mathcal{N}\}$. Since all the added events σ', σ'' and σ_τ are uncontrollable, the new subset of prohibitable events is unchange, i.e. $\tilde{\Sigma}_{hib} = \Sigma_{hib}$. Also, since σ_τ is forcible, the new subset $\tilde{\Sigma}_{for}$ of forcible events is $\tilde{\Sigma}_{for} = \Sigma_{for} \cup \{\sigma_\tau \mid \sigma \in \Sigma_{k,com,l}^{bd}, k, l \in \mathcal{N}\}$. Note that $\tilde{\Sigma}_c := \tilde{\Sigma}_{for} \dot{\cup} \{tick\}$, and $\tilde{\Sigma}_{uc} := \tilde{\Sigma} \setminus \tilde{\Sigma}_c$.

The specifications imposed on $\tilde{\mathbf{G}}$ include both the original E and the bounded/unbounded delay requirement of communication events. Thus the new specification \tilde{E} is

$$\tilde{E} = E \parallel \{L_m(\mathbf{SPEC_BCH}_d(k, \sigma, l)) \mid \sigma \in \Sigma_{k,com,l}^{bd}, k, l \in \mathcal{N}\} \\ \parallel \{L_m(\mathbf{SPEC_CH}(k, \sigma, l)) \mid \sigma \in \Sigma_{k,com,l}^{ud}, k, l \in \mathcal{N}\}$$

where “ \parallel ” denotes synchronous product [14].

As we have mentioned, a consequence of introducing the communication channels is that the agents \mathbf{G}_k ($k \in \mathcal{N}$) generally have distinct observable event sets. Hence the local preemptors/controllers to be allocated to different agents will be required to have different observable event sets. To address this, rather than synthesizing a monolithic supervisor, we propose to synthesize N decentralized supervisors one for each observable event set $\tilde{\Sigma}_{o,k}$ ($k \in \mathcal{N}$) given by

$$\tilde{\Sigma}_{o,k} := (\Sigma_o \setminus \Sigma_{com,k}) \cup \{\sigma, \sigma'', \sigma_\tau \mid \sigma \in \Sigma_{k,com,l}^{bd}, l \in \mathcal{N}, l \neq k\} \\ \cup \{\sigma, \sigma'' \mid \sigma \in \Sigma_{k,com,l}^{ub}, k, l \in \mathcal{N}, l \neq k\} \\ \cup \{\sigma' \mid \sigma \in \Sigma_{l,com,k}, l \in \mathcal{N}, l \neq k\}.$$

For the synthesis of decentralized supervisors, we employ the concept of *relative coobservability* [7]. Let $\tilde{P}_k : \tilde{\Sigma}^* \rightarrow \tilde{\Sigma}_{o,k}^*$ and $C \subseteq L_m(\tilde{\mathbf{G}})$ be an ambient language. A sublanguage $K \subseteq C$ is *relatively coobservable* (with respect to C , $\tilde{\mathbf{G}}$ and \tilde{P}_k , $k \in \mathcal{N}$), or simply C -coobservable, if for every $k \in \mathcal{N}$ and every pair of strings $s, s' \in \Sigma^*$ with $\tilde{P}_k(s) = \tilde{P}_k(s')$ there holds

$$(\forall \sigma \in \Sigma_k) \ s\sigma \in \overline{K}, s' \in C, s'\sigma \in L(\tilde{\mathbf{G}}) \Rightarrow s'\sigma \in \overline{K}.$$

Namely, relative coobservability of K requires that K be relatively observable with respect to each $\tilde{P}_{o,k}$ and Σ_k , $k \in \mathcal{N}$. It is proved [7] that relative coobservability is in general stronger than coobservability, weaker than conormality, and closed under set union. Therefore, there exists a unique supremal relatively coobservable sublanguage of a given language, which may be effectively computed.

For the new plant $\tilde{\mathbf{G}}$ and specification language \tilde{E} , write $\mathcal{CCO}(\tilde{E} \cap L_m(\tilde{\mathbf{G}}))$ for the family of relatively coobservable (and controllable, $L_m(\tilde{\mathbf{G}})$ -closed) sublanguages of $\tilde{E} \cap L_m(\tilde{\mathbf{G}})$. Then $\mathcal{CCO}(\tilde{E} \cap L_m(\tilde{\mathbf{G}}))$ is nonempty (the empty language \emptyset belongs) and has a unique supremal element

$$\sup \mathcal{CCO}(\tilde{E} \cap L_m(\tilde{\mathbf{G}})) = \bigcup \{K | K \in \mathcal{CCO}(\tilde{E} \cap L_m(\tilde{\mathbf{G}}))\}.$$

With some abuse of notation, let the generator **SUP** be such that

$$L_m(\mathbf{SUP}) := \sup \mathcal{CCO}(\tilde{E} \cap L_m(\tilde{\mathbf{G}})). \quad (21)$$

We call **SUP** the *controllable and coobservable controlled behavior*, and assume that $L_m(\mathbf{SUP}) \neq \emptyset$.¹

Next, for each observable event set $\tilde{\Sigma}_{o,k}$ ($k \in \mathcal{N}$), we construct as in (10) a *partial-observation decentralized supervisor* \mathbf{SUPO}_k defined over $\tilde{\Sigma}_{o,k}$. It is well known that such constructed decentralized supervisors \mathbf{SUPO}_k collectively achieve the same controlled behavior as **SUP** does. The control actions of the supervisor \mathbf{SUPO}_k include (1) disabling prohibitable events in

$$\tilde{\Sigma}_{hib,k} := \Sigma_k \cap \tilde{\Sigma}_{hib}. \quad (22)$$

and (2) preempting event *tick* via forcible events in

$$\tilde{\Sigma}_{for,k} := (\Sigma_k \cap \Sigma_{for}) \cup \{\sigma_\tau | \sigma \in \Sigma_{k,com,l}^{bd}, l \in \mathcal{N}\} \quad (23)$$

Note that $\tilde{\Sigma}_{for,k}$ includes not only the forcible events in $\Sigma_k \cap \Sigma_{for}$, but also the forcible events σ_τ in those bounded channel models transmitting events $\sigma \in \Sigma_k$. Also, since Σ_k are pairwise distinct, so are $\tilde{\Sigma}_{hib,k}$ and $\tilde{\Sigma}_{for,k}$, ($k \in \mathcal{N}$), and therefore $\tilde{\Sigma}_{hib} = \dot{\bigcup}_{k \in \mathcal{N}} \tilde{\Sigma}_{hib,k}$ and $\tilde{\Sigma}_{for} = \dot{\bigcup}_{k \in \mathcal{N}} \tilde{\Sigma}_{for,k}$.

Finally, we apply the localization procedure developed in Section IV to decompose, one at a time, each decentralized supervisor \mathbf{SUPO}_k , $k \in \mathcal{N}$. The result is a set of partial-observation local preemptors $\mathbf{LOC}_\alpha^P = (Y_\alpha, \Sigma_\alpha, \eta_\alpha, y_{0,\alpha}, Y_{m,\alpha})$, one for each forcible event $\alpha \in \tilde{\Sigma}_{for}$, as well as a set of local controllers $\mathbf{LOC}_\beta^C = (Y_\beta, \Sigma_\beta, \eta_\beta, y_{0,\beta}, Y_{m,\beta})$, one for each $\beta \in \tilde{\Sigma}_{hib}$. Owing to $\tilde{\Sigma}_{for} = \dot{\bigcup}_{k \in \mathcal{N}} \tilde{\Sigma}_{for,k}$ (resp. $\tilde{\Sigma}_{hib} = \dot{\bigcup}_{k \in \mathcal{N}} \tilde{\Sigma}_{hib,k}$), one local preemptor \mathbf{LOC}_α^P (resp. one local controller \mathbf{LOC}_β^C) will be allocated to precisely one agent.

The following is the main result of this section, which asserts that the collective controlled behavior of the resulting partial-observation local preemptors and local controllers is identical to that of **SUP**, thus satisfying all the imposed communication delay requirements.

¹The imposed temporal specifications of bounded/unbounded communication delay may cause $L_m(\mathbf{SUP}) = \emptyset$, which means that the delay requirements are too strong to be satisfied. In that case, we shall weaken the delay requirements by either decreasing delay bounds of bounded-delay channels or reducing the number of unbounded-delay channels, until we obtain a nonempty $L_m(\mathbf{SUP})$.

Theorem 2. *The set of partial-observation local preemptors $\{\mathbf{LOC}_\alpha^P | \alpha \in \tilde{\Sigma}_{hib}\}$ and the set of partial-observation local controllers $\{\mathbf{LOC}_\beta^C | \beta \in \tilde{\Sigma}_{hib}\}$ derived above are control equivalent to the monolithic supervisor **SUP** in with respect to the plant $\tilde{\mathbf{G}}$, i.e.*

$$L(\tilde{\mathbf{G}}) \cap L(\mathbf{LOC}) = L(\mathbf{SUP}) \quad (24)$$

$$L_m(\tilde{\mathbf{G}}) \cap L_m(\mathbf{LOC}) = L_m(\mathbf{SUP}) \quad (25)$$

with

$$\begin{aligned} L(\mathbf{LOC}) &:= \left(\bigcap_{\alpha \in \tilde{\Sigma}_{for}} P_\alpha^{-1} L(\mathbf{LOC}_\alpha^P) \right) \\ &\quad \cap \left(\bigcap_{\beta \in \tilde{\Sigma}_{hib}} P_\beta^{-1} L(\mathbf{LOC}_\beta^C) \right) \end{aligned} \quad (26)$$

$$\begin{aligned} L_m(\mathbf{LOC}) &:= \left(\bigcap_{\alpha \in \tilde{\Sigma}_{for}} P_\alpha^{-1} L_m(\mathbf{LOC}_\alpha^P) \right) \\ &\quad \cap \left(\bigcap_{\beta \in \tilde{\Sigma}_{hib}} P_\beta^{-1} L_m(\mathbf{LOC}_\beta^C) \right) \end{aligned} \quad (27)$$

where $P_\alpha : \tilde{\Sigma}^* \rightarrow \Sigma_\alpha^*$ and $P_\beta : \tilde{\Sigma}^* \rightarrow \Sigma_\beta^*$.

The proof of Theorem 2 is similar to that of Theorem 1, which relies on the facts that (1) for each forcible event, there is a corresponding partial-observation local preemptor that preempts event *tick* consistently with **SUP**, and (2) for each prohibitable event, there is a corresponding partial-observation local controller that disables/enables it consistently with **SUP**.

By the above localization approach, each agent \mathbf{G}_k ($k \in \mathcal{N}$) acquires a set of partial-observation local preemptors $\{\mathbf{LOC}_\alpha^P | \alpha \in \tilde{\Sigma}_{for,k}\}$ and a set of partial-observation local controllers $\{\mathbf{LOC}_\beta^C | \beta \in \tilde{\Sigma}_{hib,k}\}$. Thus we obtain a distributed control architecture for multi-agent TDES under partial observation and communication delay.

As described in Section IV.C, the partial-observation localization algorithm of constructing partial-observation local preemptors and controllers has the complexity exponential in $|X|$, where X is the state set of **SUP**. Introducing the communication channels will increase the state size of **SUP** and thus affects the overall complexity. Assume that there are m_1 unbounded channels $\mathbf{CH}(k, \sigma, l)$ (as displayed in Fig. 7) and m_2 bounded channels $\mathbf{BCH}(k, \sigma, l)$ (as displayed in Fig. 6) with delay bound d . Thus there are m_1 generators $\mathbf{SPEC_CH}(k, \sigma, l)$ and m_2 generators $\mathbf{SPEC_BCH}_d(k, \sigma, l)$ representing the specifications on unbounded and bounded delays respectively; the state sizes of the channels and specifications are listed in Table I. Assume that for all channels, $d \leq c_1 \in \mathbb{N}$, and the state size of the generator representing E

TABLE I. STATE SIZES OF COMMUNICATION CHANNELS AND SPECIFICATIONS

TDES channels	state size	specifications	state size
$\mathbf{CH}(k, \sigma, l)$	3	$\mathbf{SPEC_CH}(k, \sigma, l)$	2
$\mathbf{BCH}(k, \sigma, l)$	$(1/2*(d+2)(d+3))$	$\mathbf{SPEC_BCH}_d(k, \sigma, l)$	$d+2$

is c_2 . By (21), $|X| \leq 2^{|Q|*f_1*c_2*f_2}$, where $f_1 = 3^{m_1}(1/2 * (c_1 + 2)(c_1 + 3))^{m_2}$ is the states number of the composition of all communication channels, and $f_2 = 2^{m_1} * (c_1 + 2)^{m_2}$ is the states number of the synchronous product of all the specifications on the channels. So the complexity of partial-observation localization procedure under communication delay is double-exponential in $|Q| * f_1 * c_2 * f_2$. It is true that when the system is large-scale, the computation of local preemptors/controllers is impractical for general computers. So in that case, the localization procedure should be combined with some efficient heterarchical supervisory synthesis approach (e.g. [16]); we will investigate the details in future work.

Proof of Theorem 2: We provide the proof of the (\supseteq) direction of (24) and (25) may be verified analogously as in the proof of Theorem 1. Here we prove (\subseteq) of (24) by induction, i.e. $L(\tilde{\mathbf{G}}) \cap L(\mathbf{LOC}) \subseteq L(\mathbf{SUP})$.

For the **base step**, note that none of $L(\tilde{\mathbf{G}})$, $L(\mathbf{LOC})$ and $L(\mathbf{SUP})$ is empty; and thus the empty string ϵ belongs to all of them. For the **inductive step**, suppose that $s \in L(\tilde{\mathbf{G}}) \cap L(\mathbf{LOC})$, $s \in L(\mathbf{SUP})$ and $s\sigma \in L(\tilde{\mathbf{G}}) \cap L(\mathbf{LOC})$ for arbitrary event $\sigma \in \Sigma$; we must show that $s\sigma \in L(\mathbf{SUP})$. Since $\tilde{\Sigma} = \tilde{\Sigma}_{uc} \dot{\cup} \tilde{\Sigma}_{hib} \dot{\cup} \{tick\}$, we consider the following three cases.

(1) $\sigma \in \tilde{\Sigma}_{uc}$. Since $L(\mathbf{SUP})$ is controllable, and $s\sigma \in L(\tilde{\mathbf{G}})$ (i.e. $\sigma \in Elig_{\tilde{\mathbf{G}}}(s)$), we have $\sigma \in Elig_{L_m(\mathbf{SUP})}(s)$. That is, $s\sigma \in \overline{L_m(\mathbf{SUP})} = L(\mathbf{SUP})$.

(2) $\sigma = tick$. By the hypothesis that $s, s.tick \in L(\mathbf{LOC})$, for every forcible event $\alpha \in \tilde{\Sigma}_{for,k}$, $k \in \mathcal{N}$, $s, s.tick \in P_{\alpha,k}^{-1}L(\mathbf{LOC}_{\alpha,k}^P)$, i.e. $P_{\alpha,k}(s), P_{\alpha,k}(s).tick \in L(\mathbf{LOC}_{\alpha,k}^P)$. Let $y = \eta_{\alpha}(y_{0,\alpha}, P_{\alpha,k}(s))$; then $\eta_{\alpha}(y, tick)!$. The rest of the proof is similar to case (2) of proving Theorem 1, with \mathbf{LOC}_{α}^P and P_{α} replaced by $\mathbf{LOC}_{\alpha,k}^P$ and $P_{\alpha,k}$ respectively.

(3) $\sigma \in \tilde{\Sigma}_{hib}$. There must exist a partial-observation local controller $\mathbf{LOC}_{\sigma,k}^C$ for σ . It follows from $s\sigma \in L(\mathbf{LOC})$ that $s\sigma \in P_{\sigma,k}^{-1}L(\mathbf{LOC}_{\sigma,k}^C)$ and $s \in P_{\sigma,k}^{-1}L(\mathbf{LOC}_{\sigma,k}^C)$. So $P_{\sigma,k}(s\sigma) \in L(\mathbf{LOC}_{\sigma,k}^C)$ and $P_{\sigma,k}(s) \in L(\mathbf{LOC}_{\sigma,k}^C)$, namely, $\eta_{\sigma,k}(y_0, P_{\sigma,k}(s\sigma))!$ and $\eta_{\sigma,k}(y_0, P_{\sigma,k}(s))!$. Let $y := \eta_{\sigma,k}(y_0, P_{\sigma,k}(s))$; then $\eta_{\sigma,k}(y, \sigma)!$ (because $\sigma \in \Sigma_{\sigma,k}$). The rest of the proof is similar to case (3) of proving Theorem 1, with \mathbf{LOC}_{σ}^C and P_{σ} replaced by $\mathbf{LOC}_{\sigma,k}^C$ and $P_{\sigma,k}$ respectively.

□

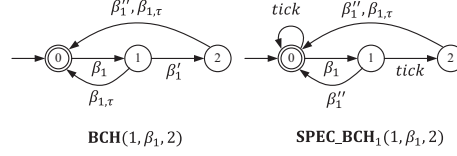


Fig. 8. Untimed DES model of bounded communication channel $\mathbf{BCH}(1, \beta_1, 2)$ and the corresponding specification $\mathbf{SPEC_BCH}_1(1, \beta, 2)$.

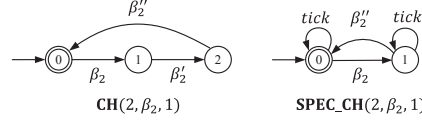


Fig. 9. Untimed DES model of unbounded communication channel $\mathbf{CH}(2, \beta_2, 1)$ and the corresponding specification $\mathbf{SPEC_CH}(2, \beta, 1)$.

For illustration, the proposed localization procedure is applied to study a timed workcell under bounded and unbounded communication delay in the following subsection.

C. Timed WorkCell Example

We demonstrate the proposed partial-observation localization with communication delay by studying the distributed control of the timed workcell example described in Section V. For illustration, we consider the case that event β_1 should be transmitted from **M1** to **M2** with delay bound 1 (*tick*), event β_2 transmitted from **M2** to **M1** with unbounded delay, and events μ_1 and η_1 are unobservable.

First, we create the bounded communication channel $\mathbf{BCH}(1, \beta_1, 2)$ to transmit event β_1 , as displayed in Fig. 8 and unbounded communication channel $\mathbf{CH}(2, \beta_2, 1)$ to transmit event β_2 , as displayed in Fig. 9. For the given delay bounds imposed on the channels, the specifications on the channels are $\mathbf{SPEC_BCH}_1(1, \beta_1, 2)$, and $\mathbf{SPEC_CH}(2, \beta_2, 1)$, as displayed in Fig. 8 and Fig. 9 respectively. The lower and upper bounds of the newly added events are enumerated in Table II.

Then, the plant to be controlled is

$$\mathbf{NPLANT} = \mathbf{Comp}(\mathbf{M1}, \mathbf{M2}, \\ \mathbf{BCH}(1, \beta_1, 2), \mathbf{CH}(2, \beta_2, 1)),$$

TABLE II. LOWER AND UPPER TIME BOUNDS OF EACH SIGNAL EVENT

event label	(lower, upper) bounds	event label	(lower, upper) bounds
β'_1	(0,1)	β'_2	(0, ∞)
β''_1	(0,1)	β''_2	(0, ∞)
$\beta_{1,\tau}$	(1,1)		

TABLE III. STATE SIZE OF PARTIAL-OBSERVATION LOCAL PREEMPTOR/CONTROLLER

Local preemptor	State size	Local controller	State size
$\text{LOC}_{\alpha_1}^P$	2	$\text{LOC}_{\alpha_1}^C$	45
$\text{LOC}_{\mu_1}^P$	1	$\text{LOC}_{\mu_1}^C$	2
$\text{LOC}_{\alpha_2}^P$	4	$\text{LOC}_{\alpha_2}^C$	12
$\text{LOC}_{\mu_2}^P$	2	$\text{LOC}_{\mu_2}^C$	1
$\text{LOC}_{\beta_{1,\tau}}^P$	1		

and the new specification is

$$\begin{aligned} \mathbf{NSPEC} = & \text{Sync}(\mathbf{BUFSPEC}, \mathbf{BRSPEC}, \\ & \mathbf{SPEC_BCH}_1(1, \beta_1, 2), \mathbf{SPEC_CH}(2, \beta_2, 1)). \end{aligned}$$

None of the newly added events are prohibitable, so the subset of prohibitable events is $\tilde{\Sigma}_{hib} = \{\alpha_1, \mu_1, \alpha_2, \mu_2\}$.

The timed out event $\beta_{1,\tau}$ is forcible and thus the subset of forcible events is changed as $\tilde{\Sigma}_{for} = \{\alpha_1, \mu_1, \alpha_2, \mu_2, \beta_{1,\tau}\}$. The subset of unobservable events for **M1** is $\tilde{\Sigma}_{uo,1} = \{\mu_1, \eta_1, \beta'_1, \beta_2, \beta''_2\}$ (the occurrence of β'_2 represents that **M1** has received the occurrence of β_2 , so β'_2 is observable to **M1**), and the subset of unobservable events for **M2** is $\tilde{\Sigma}_{uo,2} = \{\mu_1, \eta_1, \beta_1, \beta'_1, \beta_{1,\tau}, \beta'_2\}$ (β'_1 is observable to **M1**).

Next, we compute the controllable and coobservable controlled behavior **SUP** as in (21) which has 671 states.

Finally, we apply the partial-observation supervisor localization procedure presented in Section IV to construct a set of partial-observation local preemptors, one for each forcible event in $\tilde{\Sigma}_{for}$ and a set of partial-observation local controllers one for each prohibitable event in $\tilde{\Sigma}_{hib}$. The state sizes of the resulting local preemptors/controllers are displayed in Table III. It is verified that the collective controlled behavior of these local preemptors and controllers is equivalent to **SUP**. The control logics of the partial-observation local preemptors and controllers are similar to that of the full-observation local preemptors and controllers, as described in Section V.

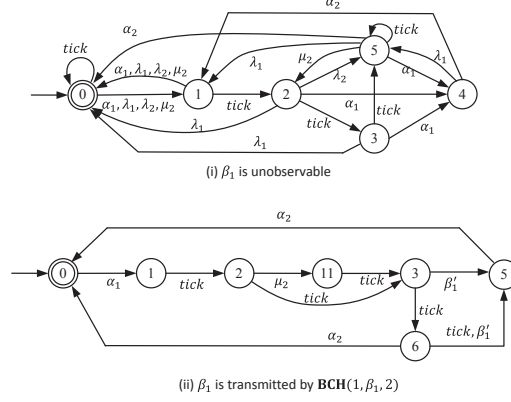


Fig. 10. Partial-observation local controller $\text{LOC}_{\alpha_2}^C$ for event α_2 . In case (i), β_1 is unobservable to $\text{LOC}_{\alpha_2}^C$ ($\Sigma_{uo} = \{\beta_1, \mu_1, \eta_1, \beta_2\}$). In case (ii), β_1 is unobservable to $\text{LOC}_{\alpha_2}^C$, but is transmitted to $\text{LOC}_{\alpha_2}^C$ by $\text{BCH}(1, \beta_1, 2)$ and thus β_1' is observable to $\text{LOC}_{\alpha_2}^C$. Here the figure of case (ii) displays part of the transition diagram of $\text{LOC}_{\alpha_2}^C$.

The event communication delays affects the control logics. For example, inspect the transition diagram of $\text{LOC}_{\mu_2}^C$; the recipient of event β_1 (represented by β_1') affects the control logics of $\text{LOC}_{\alpha_2}^C$. For illustration, consider the case that the sequence of events $\alpha_1, \text{tick}, \beta_1, \text{tick}$ has occurred, i.e., **M1** has taken a workpiece from the source (α_1) and deposited it into the buffer (β_1). Now, event α_2 (**M2** takes the workpiece from the buffer) is eligible to occur. However, since β_1 is unobservable to the local controller $\text{LOC}_{\alpha_2}^C$ for α_2 (the local controller does not know whether or not β_1 has occurred), $\text{LOC}_{\alpha_2}^C$ will disable event α_2 to prevent the underflow of the buffer (the control strategy is displayed as case (i) in Fig. 10). However, if the occurrence of β_1 is transmitted by $\text{BCH}(1, \beta_1, 2)$ in 1 (the delay bound) *ticks*, $\text{LOC}_{\alpha_2}^C$ will enable α_2 in time, as displayed in case (ii) of Fig. 10.

Finally, we allocate each partial-observation local preemptor/controller to the agent owning the corresponding forcible/prohibitible event, and thereby build a distributed control architecture for this workcell, as displayed in Fig. 11. A local preemptor/controller may observe an event generated by the agent owning it, as denoted by solid lines in Fig. 11, or imports an event by communication from other local preemptors/controllers with no communication delay, as denoted by the dashed lines, or receives the occurrence of an event through communication channels with communication delays (e.g. $\text{CH}(2, \beta_2, 1)$ and $\text{BCH}(1, \beta_1, 2)$). Since $L_m(\text{SUP}) \subseteq \tilde{E} = L_m(\text{NSPEC})$ and the collective controlled behavior of the local preemptors and controllers is equivalent to $L_m(\text{SUP})$, the system behavior satisfies the specification of communication delays on the channels.

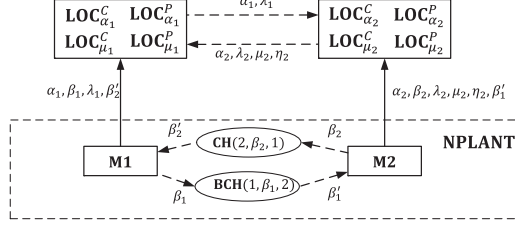


Fig. 11. Distributed control architecture with communication delay. The acknowledgements of channels are not displayed in the figure.

VI. CONCLUSIONS

In this paper, we have first developed a partial-observation supervisor localization procedure to solve the distributed control problem of multi-agent TDES. A synthesized monolithic supervisor is decomposed into a set of partial-observation local controllers and a set of partial-observation local preemptors, whose state changes are caused only by observable events. We have proved that the resulting local controllers/preemptors collectively achieve the same controlled behavior as the monolithic supervisor does.

Moreover, we have extended the partial-observation supervisor localization to the case where inter-agent event communication is subject to bounded and unbounded delay. To address communication delay, we have developed an extended localization procedure based on explicit channel models and relative coobservability. We have proved that the resulting local controllers/preemptors collectively satisfy the communication delay requirements. The above results are both illustrated by a timed workcell example.

In future research we shall extend the partial-observation localization procedure to study distributed control of large-scale systems, by combining the proposed supervisor localization with some efficient heterarchical synthesis procedure, e.g. [16].

REFERENCES

- [1] K. Cai and W. M. Wonham, "Supervisor localization: a top-down approach to distributed control of discrete-event systems," *IEEE Trans. on Automatic Control*, vol. 55, no. 3, pp. 605–618, 2010.
- [2] —, *Supervisor Localization: A Top-Down Approach to Distributed Control of Discrete-Event Systems*. Lecture Notes in Control and Information Sciences, vol. 459, Springer, 2015.
- [3] R. Zhang, K. Cai, Y. Gan, Z. Wang, and W. M. Wonham, "Supervision localization of timed discrete-event systems," *Automatica*, vol. 49, no. 9, pp. 2786–2794, 2013.
- [4] B. Brandin and W. M. Wonham, "Supervisory control of timed discrete-event systems," *IEEE Trans. on Automatic Control*, vol. 39, no. 2, pp. 329–342, 1994.

- [5] R. Zhang and K. Cai, “Supervisor localization of discrete-event systems under partial observation,” *Technical Report*, 2015, available at <http://arxiv.org/abs/1509.05498>. Also see “On supervisor localization based distributed control of discrete-event systems under partial observation”, to appear in ACC 2016.
- [6] K. Cai, R. Zhang, and W. M. Wonham, “Relative observability of discrete-event systems and its supremal sublanguages,” *IEEE Transactions on Automatic Control*, vol. 60, no. 3, pp. 659–670, 2015.
- [7] —, “Relative observability and coobservability of timed discrete-event systems,” *IEEE Transactions on Automatic Control*, published online, 2015.
- [8] F. Lin and W. M. Wonham, “Supervisory control of timed discrete-event systems under partial observation,” *IEEE Transactions on Automatic Control*, vol. 40, no. 3, pp. 558–562, 1995.
- [9] R. Zhang, K. Cai, Y. Gan, and W. M. Wonham, “Distributed supervisory control of discrete-event systems with communication delay,” *Discrete Event Dynamic Systems*, to appear, 2016.
- [10] —, “Delay-robustness in distributed control of timed discrete-event systems based on supervisor localization,” *International Journal of Control*, to appear, 2016.
- [11] G. Barrett and S. Lafortune, “Decentralized supervisory control with communicating controllers,” *IEEE Trans. on Automatic Control*, vol. 45, no. 9, pp. 1620–1638, September 2000.
- [12] S.-J. Park and K.-H. Cho, “Decentralized supervisory control of discrete event systems with communication delays based on conjunctive and permissive decision structures,” *Automatica*, vol. 43, no. 4, pp. 738–743, April 2007.
- [13] F. Lin, “Control of networked discrete event systems: dealing with communication delays and losses,” *SIAM J. Control and Optimization*, vol. 52, no. 2, pp. 1276–1298, 2014.
- [14] W. M. Wonham, *Supervisory Control of Discrete-Event Systems*. Systems Control Group, ECE Dept, Univ. Toronto, Toronto, ON, Canada, July 2015, available at <http://www.control.utoronto.ca/DES>.
- [15] R. Zhang and K. Cai, “Supervisor localization of timed discrete-event systems under partial observation and communication delay,” *Technical Report*, 2016, available at <http://arxiv.org/abs/1603.02023>.
- [16] L. Feng and W. M. Wonham, “Supervisory control architecture for discrete-event systems,” *IEEE Trans. Autom. Control*, vol. 53, no. 6, pp. 1449–1461, 2008.